IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-6, Issue-3 E-ISSN: 2347-2693

Performance Evaluation of ACO and PSO Task Scheduling Algorithms in Cloud Environment

D. Gupta[1*], H.J.S. Sidhu[2]

[1*]CSE, Desh Bhagat University, Mandi Gobindgarh, India
[2*]CSE, Desh Bhagat University, Mandi Gobindgarh, India

Available online at: www.ijcseonline.org

Received: 20/Feb//2018, Revised: 26/Feb2018, Accepted: 19/Mar/2018, Published: 30/Mar/2018

Abstract: Use of cloud technology for different requirements of an organization is on increase. Number of companies like Amazon, Microsoft, Salesforce, etc. is leading the package of cloud services. The main objective of these companies is to ensure that right resources are assigned to clients so that the resources are not left underutilized. Cloud task scheduling is a key research area and every company is investing a lot into it to reduce the underutilization of resources and ensuring the tasks finish on time. Metaheuristic algorithms over time have been used extensively for this task. This paper analyzes the performance of two metaheuristic algorithms namely ACO & PSO for cloud task scheduling.

Keywords: ACO, PSO, VM, SJF, IAAS, PAAS, SAAS, Data Centre, Cloud Computing, ETC.

I. INTRODUCTION

Cloud computing is fast emerging as an efficient and affordable means of providing platforms such as IaaS (Infrastructure as a service), SaaS (Software as a service), PaaS (Platform as a service) and much more to the service requesters. A large number of commercial applications are available that provide wide range of services over cloud. Increased and fast access to internet has also led to a boom in this sector. Different services are put within the virtualized resources of a cloud, enabling it to carry out abstractions of its underlying resources. Typically, the virtualization of a service implies the aggregation of several proprietary processes collected in a virtual environment, called Virtual Machine (VM) [1,2].

Resources for research and institutional uses have become very costly. Cloud platform lets you use these resources on rental basis[4]. You only pay for what you use and for the time you use it. For a service provider the idea is to increase the profits and ensure that the resources are utilized optimally. Meta-heuristic algorithms are particularly useful when implemented as online algorithms. All cloudlets are not passed to the load balancer or scheduler in a single go. The algorithm is not aware of the total number of cloudlets or requests. Only a set of cloudlets are passed to it that it has to schedule in one go. The other cloudlets are passed to it in second phase and so on. Static algorithms like SJF (Shortest job first) also produce good results but are not useful in case the cloudlets are passed to it in online fashion. SJF is only used in case all cloudlets are passed to it in single phase.

Section 2 & 3 of this paper gives a brief overview of both Particle Swarm Optimization and Ant Colony optimization. Section 4 discusses the methodology for implementing the task scheduling in cloud environment using both Particle Swarm Optimization and Ant Colony optimization. Section 5 discusses the results/ outcome of the two algorithms. In section 6, the paper provides a brief conclusion on the comparison of performance of two algorithms.

II. PARTICLE SWARM OPTIMIZATION (PSO)

Particle Swarm Optimization (PSO) is a swarm-based meta-heuristic algorithm influenced by the social behaviour of animals [7]. A particle in PSO is comparable to a bird or fish. PSO algorithm is based on the position of each particle in the swarm that changes with time till it finds the best solution. The position of particles in a solution space represents a solution for the problem[9]. The movement of each particle is dependent on velocity. Velocity refers to both magnitude and direction. Each particle position at any instance of time is influenced by its best position and the position of the best particle in a problem space. The performance of a particle is measured by a fitness value, which is problem specific. PSO has gained popularity due to its simplicity and its usefulness in broad range of applications with low computational cost.

PSO has fewer primitive mathematical operators than other metaheuristic algorithms which results in lesser convergence time. PSO is generally applied to continuous value problems and cloud task scheduling is a discrete problem. The first step of PSO scheduling problem is to encode the problem.

Next step is to represent the particle a particle can be a single assignment of task to a cloudlet or a vector of size n, where n is the no. of tasks and value assigned to each position is the resource index. Thus the particle represents mapping of resource to a task. Velocities are also represented in the form of vectors[3].

III. ANT COLONY OPTIMIZATION (ACO)

Ant Colony Optimization (ACO) is a meta-heuristic algorithm. It is inspired by the behaviour of real ants looking for the shortest path between their colonies and a source of food. Ants leave pheromones on their way from colony to the food source. The pheromone intensity on the passages increases with the number of ants passing through and drops with the evaporation of pheromone. As the time goes on, smaller paths draw more pheromone and thus, pheromone intensity helps ants to recognize smaller paths to the food source [6].

Ant Colony Optimization algorithm can help us in solving the cloud computing resource management and job scheduling problem. For scheduling of independent tasks in grid [6] or cloud, the number of ants taken is less than or equal to number of tasks.

IV. METHODOLOGY

Particle Swarm Optimization

Initially the tasks should be assigned to different VM's using some random algorithm. The selected algorithm can be a static or dynamic algorithm. The tasks can also be assigned randomly without the use of any task scheduling algorithm. The idea is to have all tasks assigned to a VM before the load balancing in implemented[8].

Once the task scheduling is achieved, the next step is to compute the load of all VM's. The idea is to identify which VM's are under-loaded and which VM's are over-loaded.

Then the Particle swarm optimization load balancing algorithm is used to migrate tasks from the overloaded VM's to the under-loaded VM's.

Algorithm

1. Set particle dimension as equal to the size of ready tasks T.
2. Initialize particles position randomly from PC = 1.....j and velocity vi randomly.
3. For each particle, calculate its fitness value.
4. If the fitness value is better than the previous best pbest, set the current fitness value as the new pbest.
5. Perform Steps 3 and 4 for all particles and select the best particle as gbest.
6. For all particles, calculate velocity and update their positions.
7. If the stopping criteria or maximum iteration is not satisfied, repeat from Step 3 & 4.

Ant Colony Optimization

Each ant starts with an arbitrary task ti and resource Rj for processing this task. Next, the task to be executed and the resource on which it is performed are calculated using a probable function [3]. In this way, step by step, each ant builds the whole solution of assigning all the tasks to the resources. Initially, the pheromone value is set to be a positive constant and then ants change this value at the end of every iteration. The ant's solution that gives minimum value or maximum value for considered objective function is taken as the best solution of that iteration. The final optimal solution is the one which is best out of all iteration's best solution[5].

Algorithm

1. Initialize pheromone value for each path between tasks and resources, set optimal solution to NULL and place m ants on random resources.
2. Repeat for each ant
   a. Put the starting resource of first task in tabu list and all other tasks in allowed list.
   b. Based on the probability function or transition rule, select the resource for all remaining tasks in the allowed list.
3. Compute fitness of all ants which in this case is Makespan time.
4. Replace the optimal solution with the ant's solution having best fitness value if its value of better than previous optimal solution.
5. Update both local and global pheromone.
6. Stop when the termination condition is met and print the optimal solution.

V. RESULTS

The pheromone intensity in case of ACO algorithm is minimal in the early stages of the implementation. The algorithm performs much better at latter stages and is particularly useful with large number of cloudlets. PSO on the other hand suffers from the problem of converging too fast and getting caught in local optimal. Hence its performance does not improve as drastically as compared to ACO.

Table 1: Parameters Setting of Cloudsim

Entity Type	Parameters	Values
Task(Cloudlet)	Length of Task	20000-400000
	Total Number of Task	150-250
Virtual Machine	Total Number of VMs	8
	MIPS	1024-4096
	VM Memory (RAM)	128-512
	Bandwidth	500-2000
	Cloudlet Scheduler	Time_shared and Space_shared
	Number of PEs Requirement	2
Data Centre	Number of Datacenter	1
	Number of Host	3
	VM Scheduler	Time_shared and Space_shared

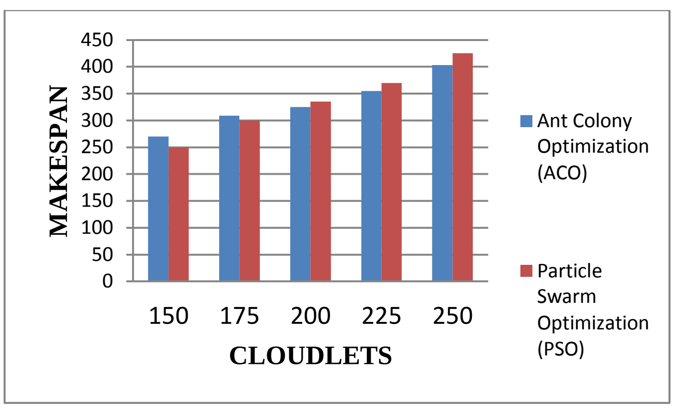

Fig.1: Makespan Time for ACO and PSO Task scheduling with fixed VMs = 8

As you can see in the figure above, performance of PSO is better than ACO with smaller number of cloudlets. One reason could be that pheromone intensity in case of ACO algorithm is minimal in the early stages and by the time algorithm makes necessary adjustments the cloudlets are over. The improvement in Makespan time of ACO is much better than that of PSO as the cloudlets are increased.

VI. CONCLUSION

Metaheuristic algorithms are meant to solve large problems that are difficult in nature and require more than polynomial time to solve. In this paper two metaheuristic problems are used to solve the task scheduling problem. ACO and PSO both tend to show similar performance except that improvement in ACO is slightly better than that of PSO. For 150 cloudlets, performance of PSO is better than that of ACO. But as the size of cloudlets is increased to 250, performance of ACO is only slightly better than that of PSO. This gap will increase if we increase the size of cloudlets even further. The difference in the finish time of last task on best and worst VMs is high. This difference is called imbalance factor. There is a need to make necessary changes into the basic structure of metaheuristic algorithms to yield even better result. It will be fruitful if a hybrid of both ACO and PSO is formed that takes advantages of the two.

REFERENCES

[1] Celesti, A., Fazio, M., Villari, M., Puliafito, A., "*Virtual machine provisioning through satellite communications in federated cloud environments*." Futur Gener Comput Syst **28**(1), 85–93 (2012)

[2] Buyya, R., Broberg, J., Goscinski, A. (eds.): "*Cloud Computing, Principles and Paradigms*". Wiley, Hoboken (2011)

[3] M. Kalra, S. Singh, "*A review of metaheuristic scheduling technique in cloud computing*", Cairo University, Egyptian Informatics Journal, Vol. 16, issue 3, pp 275-295, 2015.

[4] Anitha H M, P. Jayarekha , "*Security Challenges of Virtualization in Cloud Environment*", International Journal of Scientific Research in Computer Science and Engineering, Vol.6, Issue.1, pp.37-43, 2018..

[5] M. Tawfeek, A. El-Sisi, A. Keshk, F. Torkey, "*Cloud Task Scheduling Based on Ant Colony Optimization*" ,The International Arab Journal of Information Technology, Vol. 12, No. 2, March 2015.

[6] N. Siddique, H. Adeli, "*Nature Inspired Computing: An Overview and Some Future Directions*", Cong Comput, Vol 7, 706-714, 2015.

[7] A. Xu, Y. Yang, Z. Mi, "*Task Scheduling algorithm based on PSO in cloud environment*", IEEE 12th International Conference on Autonomic and Trusted Computing, 978-1-4673-7212-1, 2015.

[8] F. Ramezani, J. Lu, F. K. Hussain. "*Task-Based System Load Balancing in Cloud Computing Using Particle Swarm Optimization*", International Journal on Parallel Programming, Volume 42, Issue 5, pp 739–754, 2014.

[9] Shruti, M. Sharma, "*Task Scheduling and Resource Optimization in Cloud Computing using Deadline-Aware Particle Swarm Technique*", International Journal of Computer Science and Engineering", Vol. 5, Issue 6, pp. 227-231, 2017.

Authors Profile

Mr. Dinesh Gupta, did his M. Tech in IT from Department of CSE GNDU Amritsar, India. Currently he is pursuing his Ph. D in CSE from Desh Bhagat University. He has more than 8 years of experience in teaching. Currently he is working as Assistant Professor in department of CSE, IKGPTU, India. He has more than 8 publications in leading research Journal.

Dr. Harmaninder J. S. Sidhu is serving as Assis-tant Professor in Department of Computer Science and Applications, Desh Bhagat Uni-versity, Mandi Gobindgarh, Punjab, India. He is involved in academic activities for last 10 years. His interest areas include Open Source Software and Algorithm Analysis.

